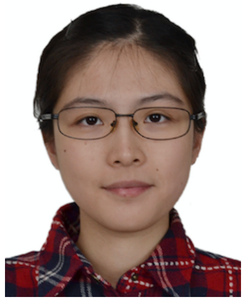
Complexity of Highly Parallel Non-Smooth Convex Optimization

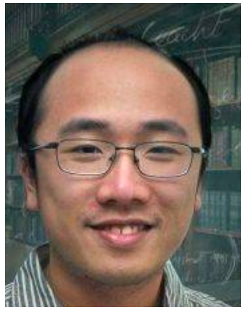
NeurIPS 2019 Spotlight

joint work with

Sébastien Bubeck



Qijia Jiang



Yin Tat Lee

Yuanzhi Li

Aaron Sidford

Non-smooth Convex Optimization

Query: $x \in \mathbb{R}^d$

$\downarrow$

First Order Oracle

$\downarrow$

$f(x), \nabla f(x)$

$$\min_{x \in \mathbb{R}^d} f(x)$$

Assumptions

- f is convex
- f is 1-Lipschitz, $|f(y) - f(x)| \leq \|y - x\|$
- $OPT = f(x_*)$ where $\|x_*\| \leq 1$

Non-smooth Convex Optimization

Query: $x \in \mathbb{R}^d$

$\downarrow$

First Order Oracle

$\downarrow$

$f(x), \nabla f(x)$

$$\min_{x \in \mathbb{R}^d} f(x)$$

Assumptions

- f is convex
- f is 1-Lipschitz, $|f(y) - f(x)| \leq \|y - x\|$
- $OPT = f(x_*)$ where $\|x_*\| \leq 1$

Goal

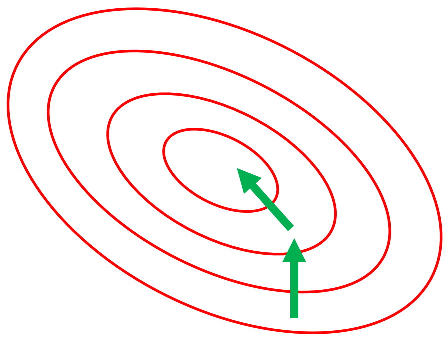
- Compute "ϵ-optimal point": $f(x) \leq OPT + \epsilon$
- Use as few queries as possible

Algorithms

- **Goal**: ϵ-optimal point for convex f
- **Oracle**: first order
- **Assumptions**: 1-Lipschitz, $\|x_*\|_2 \leq 1$

(Sub)-Gradient Descent

- $x_{k+1} = x_k - \eta \nabla f(x_k)$
- Output average $\bar{x}_k = \frac{1}{k} \sum x_k$
- $O(1/\epsilon^2)$ queries suffice

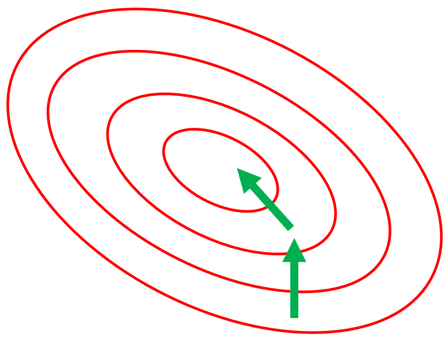


Algorithms

- **Goal**: ϵ-optimal point for convex f
- **Oracle**: first order
- **Assumptions**: 1-Lipschitz, $\|x_*\|_2 \leq 1$

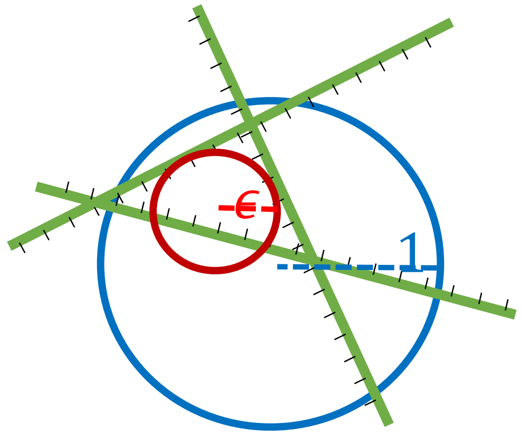
(Sub)-Gradient Descent

- $x_{k+1} = x_k - \eta \nabla f(x_k)$
- Output average $\bar{x}_k = \frac{1}{k} \sum x_k$
- $O(1/\epsilon^2)$ queries suffice



Cutting Plane Methods

- Center of gravity / high dimensional binary search
- $O(d \log(1/\epsilon))$ queries suffice

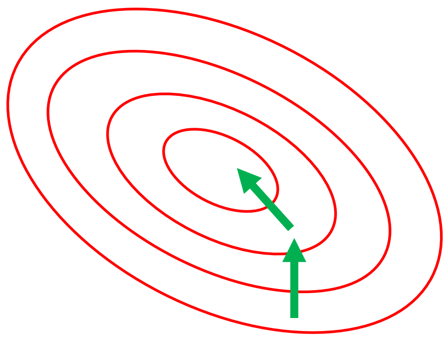


Algorithms

- **Goal**: ϵ-optimal point for convex f
- **Oracle**: first order
- **Assumptions**: 1-Lipschitz, $\|x_*\|_2 \leq 1$

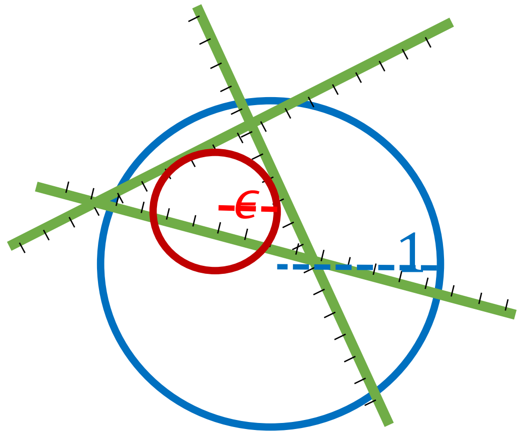
(Sub)-Gradient Descent

- $x_{k+1} = x_k - \eta \nabla f(x_k)$
- Output average $\bar{x}_k = \frac{1}{k} \sum x_k$
- $O(1/\epsilon^2)$ queries suffice



Cutting Plane Methods

- Center of gravity / high dimensional binary search
- $O(d \log(1/\epsilon))$ queries suffice



Optimal?

Algorithms

- **Goal**: ϵ-optimal point for convex f
- **Oracle**: first order
- **Assumptions**: 1-Lipschitz, $\|x_*\|_2 \leq 1$

(Sub)-Gradient Descent

- $x_{k+1} = x_k - \eta \nabla f(x_k)$
- Output average $\bar{x}_k = \frac{1}{k} \sum x_k$
- $O(1/\epsilon^2)$ queries suffice

Cutting Plane Methods

- Center of gravity / high dimensional binary search
- $O(d \log(1/\epsilon))$ queries suffice

Lower Bound

Unimprovable when

$\epsilon = \omega(1/\sqrt{d})$

Lower Bound

Unimprovable when

$\epsilon = O(1/\sqrt{d})$

Algorithms

Parallelizable?

- **Goal**: ϵ-optimal point for convex f
- **Oracle**: first order
- **Assumptions**: 1-Lipschitz, $\|x_*\|_2 \leq 1$

(Sub)-Gradient Descent

- $x_{k+1} = x_k - \eta \nabla f(x_k)$
- Output average $\bar{x}_k = \frac{1}{k} \sum x_k$
- $O(1/\epsilon^2)$ queries suffice

Cutting Plane Methods

- Center of gravity / high dimensional binary search
- $O(d \log(1/\epsilon))$ queries suffice

Lower Bound

Unimprovable when

$\epsilon = \omega(1/\sqrt{d})$

Lower Bound

Unimprovable when

$\epsilon = O(1/\sqrt{d})$

[Nemirovski 1994]

Parallel Non-smooth Convex Optimization

Query: $x \in \mathbb{R}^d$

First Order Oracle

$f(x), \nabla f(x)$

$$\min_{x \in \mathbb{R}^d} f(x)$$

Assumptions

- f is convex and 1-Lipschitz
- $OPT = f(x_*)$ for $\|x_*\| \leq 1$

Goal

- Compute ϵ-optimal point

[Nemirovski 1994]

Parallel Non-smooth Convex Optimization

Query: $x \in \mathbb{R}^d$

$\downarrow$

First Order Oracle

$\downarrow$

$f(x), \nabla f(x)$

$$\min_{x \in \mathbb{R}^d} f(x)$$

Query: $x_1, \dots, x_k \in \mathbb{R}^d$

$\downarrow$

Parallel First Order Oracle

$\downarrow$

$f(x_1), \nabla f(x_1), \dots, f(x_k), \nabla f(x_k)$

Assumptions

- f is convex and 1-Lipschitz
- $OPT = f(x_*)$ for $\|x_*\| \leq 1$

Goal

- Compute ϵ-optimal point

[Nemirovski 1994]

Parallel Non-smooth Convex Optimization

Query: $x \in \mathbb{R}^d$

First Order Oracle

$f(x), \nabla f(x)$

$$\min_{x \in \mathbb{R}^d} f(x)$$

Query: $x_1, \ldots, x_k \in \mathbb{R}^d$

Parallel First Order Oracle

$f(x_1), \nabla f(x_1), \ldots, f(x_k), \nabla f(x_k)$

Assumptions

- f is convex and 1-Lipschitz
- $OPT = f(x_*)$ for $\|x_*\| \leq 1$

Goal

- Compute ϵ-optimal point

Parallel Complexity

- **Depth**: # queries to parallel oracle
- **Work**: # gradients computed / functions evaluated

[Nemirovski 1994]

Parallel Non-smooth Convex Optimization

Query: $x \in \mathbb{R}^d$

First Order Oracle

$f(x), \nabla f(x)$

$$\min_{x \in \mathbb{R}^d} f(x)$$

Query: $x_1, \ldots, x_k \in \mathbb{R}^d$

Parallel First Order Oracle

$f(x_1), \nabla f(x_1), \ldots, f(x_k), \nabla f(x_k)$

Assumptions

- f is convex and 1-Lipschitz
- $OPT = f(x_*)$ for $\|x_*\| \leq 1$

Goal

- Compute ϵ-optimal point

Parallel Complexity

- **Depth**: # queries to parallel oracle
- **Work**: # gradients computed / functions evaluated

Our Work

- Focus on "highly parallel setting"
- $k = \text{poly}(d)$, work $= \text{poly}(d)$
- **Question**: best possible depth?

State-of-the-Art

- **Goal**: ϵ-optimal point for convex f
- **Oracle**: highly parallel first order
- **Assumptions**: 1-Lipschitz, $\|x_*\|_2 \leq 1$

(Sub)-Gradient Descent

Depth $O(1/\epsilon^2)$

Cutting Plane Methods

Depth $O(d \log(1/\epsilon))$

State-of-the-Art

- **Goal**: ϵ-optimal point for convex f
- **Oracle**: highly parallel first order
- **Assumptions**: 1-Lipschitz, $\|x_*\|_2 \leq 1$

(Sub)-Gradient Descent

Depth $O(1/\epsilon^2)$

Accelerated stochastic method

[DBW12]

Depth $O(d^{1/4}/\epsilon)$

Improves when
$\epsilon \in [d^{-3/4}, d^{-1/4}]$
depth $\in [\sqrt{d}, d]$

Cutting Plane Methods

Depth $O(d \log(1/\epsilon))$

State-of-the-Art

- **Goal**: ϵ-optimal point for convex f
- **Oracle**: highly parallel first order
- **Assumptions**: 1-Lipschitz, $\|x_*\|_2 \leq 1$

(Sub)-Gradient Descent

Depth $O(1/\epsilon^2)$

Lower Bound[N94,BS18]

No randomized algorithm improves when $\epsilon = \widetilde{\omega}(d^{-1/6})$, depth $= \tilde{O}(d^{1/3})$

Accelerated stochastic method

[DBW12]

Depth $O(d^{1/4}/\epsilon)$

Improves when
$\epsilon \in [d^{-3/4}, d^{-1/4}]$
depth $\in [\sqrt{d}, d]$

Cutting Plane Methods

Depth $O(d \log(1/\epsilon))$

State-of-the-Art

- **Goal**: ϵ-optimal point for convex f
- **Oracle**: highly parallel first order
- **Assumptions**: 1-Lipschitz, $\|x_*\|_2 \le 1$

(Sub)-Gradient Descent

Depth $O(1/\epsilon^2)$

↑

Lower Bound[N94,BS18]

No randomized algorithm improves when $\epsilon = \tilde{\omega}(d^{-1/6})$, depth $= \tilde{O}(d^{1/3})$

↑

Our Result

Improve to $\epsilon = \tilde{\omega}(d^{-1/4})$, depth $= \tilde{O}(\sqrt{d})$

Accelerated stochastic method

[DBW12]

Depth $O(d^{1/4}/\epsilon)$

↑

Improves when $\epsilon \in [d^{-3/4}, d^{-1/4}]$ depth $\in [\sqrt{d}, d]$

Cutting Plane Methods

Depth $O(d \log(1/\epsilon))$

State-of-the-Art

- **Goal**: ϵ-optimal point for convex f
- **Oracle**: highly parallel first order
- **Assumptions**: 1-Lipschitz, $\|x_*\|_2 \leq 1$

(Sub)-Gradient Descent

Depth $O(1/\epsilon^2)$

↑

Lower Bound[N94,BS18]

No randomized algorithm improves when $\epsilon = \widetilde{\omega}(d^{-1/6})$, depth $= \tilde{O}(d^{1/3})$

↑

Our Result

Improve to $\epsilon = \widetilde{\omega}(d^{-1/4})$, depth $= \tilde{O}(\sqrt{d})$

Accelerated stochastic method

[DBW12]

Depth $\mathrm{O}(d^{1/4}/\epsilon)$

↑

Improves when $\epsilon \in [d^{-3/4}, d^{-1/4}]$ depth $\in [\sqrt{d}, d]$

Improves when $\epsilon \in [d^{-1}, d^{-1/4}]$ depth $\in [\sqrt{d}, d]$ →

Cutting Plane Methods

Depth $O(d \log(1/\epsilon))$

High-order accelerated stochastic method

Our Result

Depth $\widetilde{\mathrm{O}}(d^{1/3}/\epsilon^{2/3})$

Key Takeaways

- **Goal**: ϵ-optimal point for convex f
- **Oracle**: highly parallel first order
- **Assumptions**: 1-Lipschitz, $\|x_*\|_2 \leq 1$

Lower Bound

- Gradient descent is highly-parallel optimal up to depth $\tilde{O}(\sqrt{d})$
- Previous bound was $\tilde{O}(d^{1/3})$ and ours is nearly optimal

Upper Bound

- Can improve on cutting plane whenever $\epsilon = o(d^{-1})$
- Previous bound: $\epsilon = o(d^{-3/4})$

How?

Lower Bound

- Start with [N94,BS18] instance
- Control queries of “large” norm vectors from leaking information
- Build a “wall” to shield information in lower bound from such queries

How?

Lower Bound

- Start with [N94,BS18] instance
- Control queries of “large” norm vectors from leaking information
- Build a “wall” to shield information in lower bound from such queries

Upper Bound

- Minimize convolution of f with Gaussian as in [DBW12]
- Apply high-order acceleration [GDGVSUJWZBJLL**S**19] using that can build Taylor approximation in depth 1
- Improve by broader acceleration framework and better local model than Taylor approximation

How?

Lower Bound

- Start with [N94,BS18] instance
- Control queries of “large” norm vectors from leaking information
- Build a “wall” to shield information in lower bound from such queries

Upper Bound

- Minimize convolution of f with Gaussian as in [DBW12]
- Apply high-order acceleration [GDGVSUJWZBJLL**S**19] using that can build Taylor approximation in depth 1
- Improve by broader acceleration framework and better local model than Taylor approximation

Takeaway

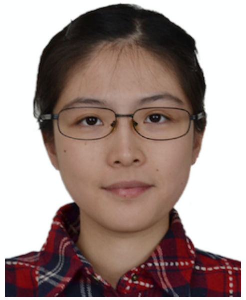
- Shielding / wall building

Takeaway

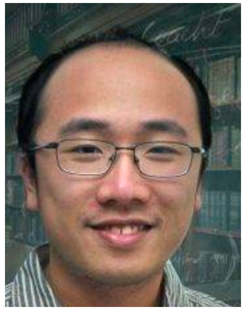
- General higher-order acceleration
- Stochastic approximation

Thank you

Sébastien Bubeck



Qijia Jiang



Yin Tat Lee

Yuanzhi Li

Aaron Sidford

Questions?

- **poster**: 5:30PM - 7:30PM @ East Exhibition Hall B + C #107
- **arXiv**: 1906.10655
- **email**: sidford@stanford.edu